Consideration of Receiver Interest for IP Multicast Delivery

Brian Neil Levine♮∗ Jon Crowcroft⋆ Christophe Diot‡∗ J.J. Garcia-Luna-Aceves† James F. Kurose♮

♮Computer Science Dept., Univ. of Massachusetts, Amherst, MA, *brian,kurose@cs.umass.edu*
†Computer Engineering Dept., Univ. of California, Santa Cruz, CA, *jj@cse.ucsc.edu*
⋆Computer Science Dept., Univ. College London, London, U.K., *jcrowcroft@cs.ucl.ac.uk*
‡Sprint Advanced Technologies Lab, Burlingame, CA, *cdiot@sprintlabs.com*

Abstract—**Large-scale applications are characterized by a large number of dynamic and often interactive group members. The nature of these applications is such that participants are not interested in all the content transmitted. We examine three currently available techniques to scope delivery of content to interested receivers in IP multicast: filtering, where data is filtered by middleware before passed to the application; addressing, where data is routed only to those receivers that express their interest; and hybrid approaches. We propose a framework that models large-scale application behavior. We use this framework to evaluate the performance of these applications and related protocols when the network is capable of filtering or addressing. Our results show that the current Internet architecture does not efficiently support large-scale applications because it can not efficiently manage multiple multicast groups. We show that network-level addressing is preferred to filtering and hybrid approaches given that groups are easy to create and manage. We highlight areas of research in the multicast architecture to bring about this change.**

I. Introduction

The complexity of networked applications is steadily increasing. Today, Internet applications manage a large and widely-dispersed set of users, have multiple data streams that vary in content and media type, and make use of multiple unicast and multicast streams in a single session. Examples of these distributed, interactive applications include multiplayer games [1], collaborative visualization and conferencing tools [2], and distributed interactive simulations (DIS) [3]. We refer to applications with a large number of users, application entities, or content and media flows as *large-scale* applications. Additionally, congestion control protocols [4], [5], reliable multicast protocols [6], [7], [8], and other multicast related end-to-end protocols [9] employ multiple unicast or multicast streams simultaneously for large receiver sets.

In this paper, we show the limits of the current IP multicast service and architecture for supporting large-scale multicast applications by examining the characteristics, requirements, and performance of such applications and related protocols. Application and protocol trends include:

- Sparse receiver interest in transmitted data.
- Partitioning of receiver sets into groups based on network performance or receiver interest in application content.
- The need for fast joins to numerous multicast groups.

Meanwhile, the trends in IP multicast network support have included support for broadcasting of data to only all group members, concealment of network performance state and topology, and architectures with high overhead. We introduce a framework to model the characteristics of large-scale applications and protocols and their network support, which we call the *working set* framework. Using this framework and simulation, we provide broad statements of what these applications and protocols require from the network for efficient performance, and compare how they are traditionally supported. We show that the current IP architecture is not prepared to support large-scale applications because it can not efficiently manage multiple multicast groups. We define a design space for future multicast multimedia applications and highlight future areas of research to bring about this change.

This paper is organized as follows. Section II reviews the requirements of different large-scale applications and transport protocols, as well as routing support options. Section III introduces our working-set model that frames the relationship between the application, transport, and network levels with regard to large-scale applications. Section IV introduces our simulation model. Transmission efficiency is compared for filtered and addressed architectures, given that application data flows are short-term or long-term. We show in Section VI why the current IP multicast service and architecture do not efficiently support large-scale applications and related transmission control protocols. Section VII presents concluding remarks and directions for future success of multicast deployment.

II. Background

Group-communication applications have different needs as compared to point-to-point applications. In this section, we review numerous large-scale applications, transport protocols, and routing mechanisms and enumerate their differences. In subsequent sections, we review how well these requirements are supported by the Internet.

A. Applications and Transport Services

Various multicast applications and transmission control protocols have a common requirement of content-based transmission. Due to their diversity of group members and data flow types, they would all benefit from an efficient group management infrastructure.

A.1 USENET News Distribution

The USENET network of NNTP (Network News Transport Protocol) [10] servers is one of the oldest and most popular large-scale Internet services. With USENET, users can post articles that are accessible by any user connected to another USENET server. USENET is interesting for our study because

† Work supported in part by the Defense Advanced Research Projects Agency (DARPA) under grants F19628-96-C-0038 and F30602-97-1-0291. ∗ Work supported in part by INRIA, Sophia Antipolis, France, and Sprint ATL, Burlingame, CA.

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2000		**00-00-2000 to 00-00-2000**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Consideration of Receiver Interest for IP Multicast Delivery	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**10**	

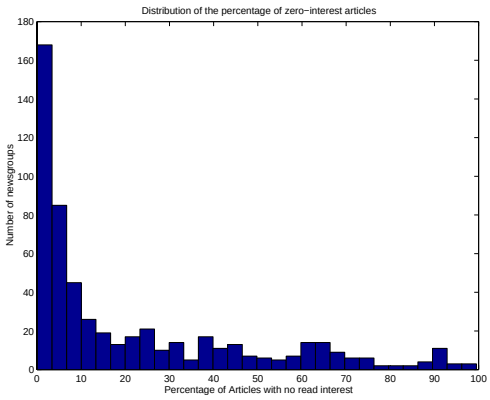


Fig. 1. Distribution of interest value of newsgroups.

it is characterized by a large set of content flows, a large and distributed user base, and a reliance on a *broadcast and filter* delivery model. Whether any users find articles useful or interesting is not considered: servers must download and store all articles posted. Users then hand filter articles. Although USENET is not multicast-based, it emulates multicast transmission and remains a good example of the problem addressed in this paper.

As motivation for our study, we recorded all requested articles at the UC Santa Cruz (UCSC) campus-wide USENET server over a period of five days[1]. Our goal was to determine what percentage of the articles posted to newsgroups were interesting to the server's readership.

USENET supports an overwhelming amount of content. The UCSC server stored articles from 7588 newsgroups with almost four million articles posted in roughly a single month. The server we studied has 20 gigabytes of disk storage, but, like most servers, still does not have the resources to subscribe to all USENET newsgroups.

Because archived material on the server was most likely already read, we focused on articles posted during the first four days of the trace and the one day previous. For each article, we determined if it was read once or more during the five-day period by any user. For each newsgroup, we calculated the amount of data in bytes that was posted during the trace, and then calculated the amount of data in bytes that was actually read. Although the server stored articles from 7588 newsgroups, during the trace only 1059 newsgroups were accessed by users, and only 662 were accessed for new articles. A histogram of the percentage of articles read for the 662 newsgroups in shown in Figure 1. For 52% of the 662 newsgroups, less than 10% of the articles were read by any user. Consequently, all but 125 megabytes of the 711 megabytes of new articles in accessed newsgroups were downloaded and stored by the server for no purpose.

Users were not interested in all or even most articles. We conjecture that the trend for content-based applications in general is that *as the number of receivers increases and the number of flows of content increases, the amount of content that is of general interest will decrease.* Clearly, the filtering mechanisms employed by USENET is extremely inefficient and without consideration to user interest in the delivered content. If USENET was designed so that USENET servers received only the articles from newsgroups that had a history of any receiver interest, then the server we studied could reduce storage requirements and network bandwidth required to transport articles by more than 80%.

While this study is not representative of every large-scale application, it illustrates a trend that is sure to appear in other content-based applications: content that is broadcast without regard to receiver interest is inefficient.

[1]User names were not recorded in the logs or seen by the authors[11].

A.2 DIS and Distributed Games

The Distributed Interactive Simulation (DIS) and the High Level Architecture (HLA) standards were designed to support networked virtual world environments. A good review of the special requirements of DIS and HLA applications is given by Pullen et al [3], and include:

- Fast leaves and joins of multicast groups. Joins should take under a second at the rate of hundreds per second.
- Thousands of simultaneous multicast groups.
- Reliability to ensure low packet loss.

One solution often used to deal with these requirements and others, including reservation of bandwidth and differentiated services, is to aggregate the contents of many multicast groups into a single multicast transmission [3]. Although DIS is real-time, the broadcast and filter network support is the same as USENET. Moreover, DIS is sure to have disparate receiver interest in simulation events because not all players in the virtual worlds can view all events.

Multiplayer games have many of the same requirements of DIS and HLA applications. Games may take on a larger number of concurrent participants than government sponsored simulations [3].

An interesting area of distributed games are serverless games, e.g., MiMaze [1]. In MiMaze, participants multicast to each other their current application state and are able to synchronize without a centralized server. The success of this mechanism is a function of the latency between participants, which is dependent on the multicast topology connecting members. Large latencies requires the group be broken up into divisions. Partitioning of group members is a strong trend in large-scale applications. Here, latency is the basis for partitioning; for USENET, DIS, and games, partitioning based on content would increase efficient data distribution if data could be *addressed* towards interested hosts, as we discuss in Section V.

Researchers in the area of distributed simulation have been concerned with the problem of interest management for a number of years [12]. Several researchers have proposed to include a specific network agent that manages the mapping of interest groups to multicast groups [13], [14]. Typically, interest grouping is done on the basis of (x,y) grid-coordinates [15], [16], a natural interest clustering for the application area of distributed interactive simulation.

A.3 Reliable Multicast

Reliable multicast protocols ensure delivery of all data transmitted to a set of multicast receivers. These protocols must manage heterogeneous receiver packet loss [17]. A solution to this

problem has been to localize feedback and recovery [7], [18], [19], which also increases scalability.

Localizing recovery traffic requires that data be multicast only to receivers that have lost the original data. Fortunately, packets lost on a multicast tree are lost on the entire subtree. Experiments have shown that receivers can be partitioned into groups based on their topology as a predictor of their common packet-loss correlation [17]. Once the groups are formed, local feedback is possible, and *subcasting*[2] can be used as an addressing mechanism to localize retransmissions without forming a new multicast group. Accordingly, partitioning is present in reliable multicast protocols based on performance metrics such as topology [17], [20] or packet loss [21].

However, IP does not support partitioning and addressing. In the case of reliable multicast protocols, complicated host-based protocols for discovering common loss patterns [21] can be used; or host-based tracing mechanisms can be used [17], [20].

A.4 Congestion Control

Multicast congestion control protocols can also exhibit receiver partitioning trends. Some protocols employ *layered encodings* of data that divide streaming data into a base encoding and multiple refining streams that enhance the rendering of the base encoding [22]. A receiver may chose to subscribe to the base-level stream, or if more bandwidth is available, to additional streams. Such protocols must discover what bandwidth is available to each receiver, and adjust the number of streams received accordingly. Receivers are partitioned based on their available bandwidth.

Lossy layered encodings are not applicable to all types of data, such as bulk file transfers. To manage a heterogeneous receiver set for bulk data transfer, protocols can send multiple data flows at heterogeneous transmission rates, where receivers subscribe to enough groups to maximize their available bandwidth usage [4]. In this scenario, some data is transmitted more than once by the source on different multicast groups, but not received twice by any receiver. The mean time of each receiver completing the data transfer is minimized.

B. Data Delivery

Data delivery support for group communication applications can be realized by *filtering* and *addressing* mechanisms. *Addressing* is a network-level or end-to-end approach to directing and routing traffic efficiently based on application-dependent content labels. *Filtering* is an approach to dropping undesired content received from the network before it is passed to the application. Having a prepared set of organized application entities and data flows allows applications to disseminate data to interested receivers only, using either addressing or filtering. We refer to the process of placing application entities and data flows into ordered structures based on their relationships in the context of the applications as *naming*.

Application Level Framing (ALF) [23] was proposed to increase the expressibility of an application's needs to lower levels. ALF architectures preserve the semantics of the application data from the sending entity to the receiving entities and all network levels in between. The ALF approach advocates that the application manage data packaging in *application data units (ADU)*, which then become the sole unit of processing, transmission, and control across all network levels.

ALF applications cannot rely on a simple, monotonically-increasing sequence space to distinguish interleaved ADUs. They require an alternate method of naming data [24], [25]. While data naming was introduced as a necessary component of ALF-based applications, we take a more general view of naming: any large-scale application requires a good organization of its application entities and content for network communication when supported by addressing or filtering.

B.1 Addressing Architectures

The IP multicast architecture is based on unique identifiers (the class D address) that lack a meaning or context at the network level and higher levels. Each multicast address is a name that identifies the group, but provides no information about the location of the participating hosts. Because multicast addresses serve only as identifiers, they would be more accurately termed "names" [26]. IP multicast does not support transport among subsets of receivers joined to an existing multicast group; packets transmitted over a multicast group can be sent only to all participants. And, there exists little or no cooperation among applications on dividing the address space so that *collisions* do not occur; applications pick multicast addresses randomly, though *session directory* applications such as `sd` and `sdr` attempt to manage the situation. For applications that require communication among subsets of hosts in the session, multiple multicast groups or multiple unicast connections are required. A good example of this approach are the many layered-multicast protocols used for multicast congestion control [4], [22].

A different approach in managing communication among hosts in a session is offered by addressable routing architectures. Recent work on addressing, such as AIM [18] enables sources to restrict the delivery of packets to a small subset of the receivers in a multicast group on a per-packet basis. AIM also permits receivers to listen to subsets of sources using lightweight multicast groups on a subscription basis, and provides routing for resource discovery, i.e., anycast routing. Similarly, PGM [19] and LWM [27] use addressable subgroups to provide a reliable multicast service. The subgroups are used to direct retransmissions of lost multicast data towards only receivers that require the missing data.

Addressable routing services, such as AIM, PGM, and LWM, rely on the application to separate data flows and to tag packet headers appropriately, i.e., naming. For complex applications, a robust structure and naming approach is required so that network-level addressing services may operate independently from any end-to-end protocol.

B.2 Application-level Filtering

Not all applications rely on multiple multicast groups or addressable routing architectures to scope data to interested receivers. Instead, a simpler, lower latency, but less efficient, approach is taken: all data is *broadcast* to all receivers using a common multicast group and receivers *filter* out packets based

[2] *Subcast* packets are multicast starting from a router other than the source and then forwarded to the remaining subtree.

on the data content. This makes sense because of the very limited deployment of multicast.

We have discussed above the use of filtering for USENET and DIS. Other examples include *Rendezvous* [28] and *Keryx* [29], which are both commercial middleware products that provide content-based filtering of broadcast data for network applications. In these schemes, all packets contain meta-information about data content. End-hosts drop received data based on the meta-information before it reaches the application. (Rendezvous is not necessarily pure broadcast, as the scheme allows for routers to forward content from very broad categories to interested hosts.)

The main disadvantage of the broadcast & filter approach (*filtering* for short) is its drain on network and end-host resources. Filtering transmissions may congest networks, fill queues at end hosts, and waste processing. Furthermore, sometimes policy, security, or privacy considerations requires that data not be broadcast to a large population of users. The primary advantages of filtering is simplicity and low latency. The responsibility of determining the interest of a received message is placed on the receiver set, and furthermore, sources of information need not know the location of receivers on the network. Another advantage of filtering is low latency because an existing multicast group is used to send data regardless of the intended receivers. As with end-to-end protocols and applications that employ addressing, the use of filtering mechanisms requires a strong naming scheme for a fast method of classifying the content.

C. Discussion

To summarize, the major characterization of large-scale applications is that there is a lack of general interest in flows of data, whether the division is by content, data encoding, transmission rate, or media type. We conjectured that as the number of receivers increases and the number of flows of data increases, the amount of flows of data that is of general interest will decrease. This results in the following requirements for large-scale applications and transport protocols, though no one exhibits all trends:

- A partitioning of receiver sets by interest in content.
- A partitioning of receivers based on performance: perceived latency, packet loss, topology, or available bandwidth.
- A need for fast joins and leaves of multicast groups.
- A need for an enormous number of concurrent multicast groups.

The trends for data delivery supporting such applications and transport protocols include

- Broadcast and filtering, where mechanisms discard undesired information.
- Addressing, where data is sent only to interested users.
- Hybrid addressing and filtering, by placing data in broad categories. Categories are addressed to interested receivers, and content within categories is filtered.

Each type of support correctly services the application: all data reaches intended receivers, ignoring packet loss. It should be apparent that addressed applications more efficiently service the application. However, most applications and transport protocols rely on a broadcast model, and a growing set rely on a hybrid approach [30]. In the following section, we introduce a framework to discuss these design options. In the subsequent sections, we model and simulate the support choices to show the network characteristics of each scenario. Then, we discuss why the Internet is not prepared to support the addressing model though it is better suited for such applications.

III. A Working Set Framework

For applications in which the data rate of the source can overwhelm either the receiving hosts or routers in between, flow and congestion control protocols traditionally have been used to limit the amount of data hosts and routers need to process. The group communication scenario introduces a new dimension along which to reduce the amount of traffic that arrives at receivers and that traverses routers: the interest value of data.

Designing efficient support for large-scale applications touches on issues involving application, transport, and network levels. The relationship between these three levels can be modeled by considering what each level manages. From this viewpoint, we can see the relationship between naming, addressing, and filtering.

Applications control a mapping between *application entities*, *data flows*, and *communication groups*:

- *Application entities* include anything that is manipulated by users in the context of the application: e.g., avatars, tanks, cameras, pointers.
- *Data flows* are a grouping by content or media type of a sequence of data transmitted between application entities. e.g., one of multiple video streams, the communication among a subgroup users, or a newsgroup in USENET.
- *Communication groups* are the set of unicast, multicast, or addressable subgroups (see Section II) used to isolate data flows. From the application perspective, communication groups are pipes transmitting data flows between application entities.

From the network perspective, there exists two manageable objects: *communication groups* and *hosts*. The management of communication groups has already been discussed. From the network point-of-view, communication groups are pipes that link hosts exchanging data. *Hosts* are simply the IP addresses of the end-users involved in the session and are the home of one or more application entities. For example, a single host with a single IP address may be in control of a few tanks in a distributed interactive simulation. From the application perspective the two tanks are separate entities, but from the network perspective, only the hosts are important.

We refer to the *working set* of a level as the set of *objects* it must manage to perform its assigned *task*.

- For the application level, the working set consists of application entities, data flows, and communication groups; the task consists of transporting data flows among application entities via communication groups.
- For the network level, the working set consists of communication groups and hosts; the task consists of transporting data among hosts via communication groups.

Figures 2, 3, and 4 illustrates how the objects of each level map together under each of the three network level scenarios. Figure 2 illustrates the relationship between application objects and network objects with an example drawn from simulation applications. Three tanks are fighting on a battlefield, and need to

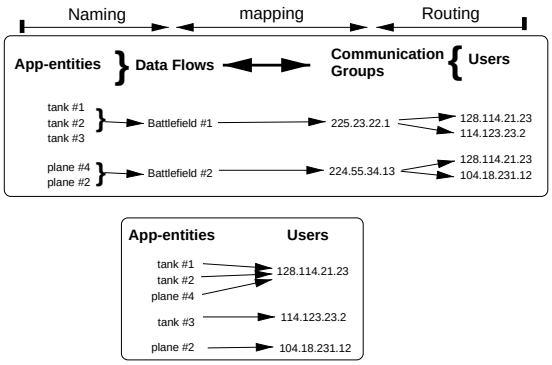


Fig. 2. Application to IP networks mapping.

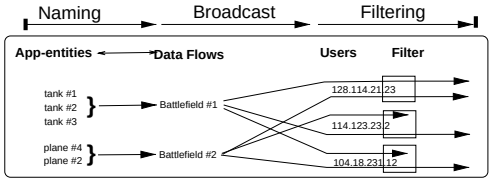


Fig. 3. Application to network mapping when over filtering.

communicate their respective movements to each other. For this purpose, the data flow "battlefield #1" connects the three tanks. The battlefield's data flow is mapped to the multicast group 225.23.22.1. The network level takes all data sent on this multicast group and disseminates it to the subscribed hosts on unicast addresses 129.103.21.23 and 114.123.23.2. At a higher level, we see a user at host 129.103.21.23 as the source of tanks 1 and 2, and 114.123.23.2 as the source of tank 3. Note that for the network level, only the communication group is important, not the associated data flows.

From this model, we can see the need for mapping data between the application and the network groupings so that the set of communication addresses used to transmit data flows between application entities arrives at, ideally, the exact corresponding sets of users, which is the case when communication groups have a one-to-one mapping with data flows[3]. When supported by IP, naming takes application entities and maps them to data flows; these data flows are mapped to communication groups, which a routing service then maps to end hosts. To ensure a one-to-one correspondence between the data flows and communication groups, IP multicast requires the maintenance of one multicast group per data flow. When there is no one-to-one mapping between data flows and communication groups, then some data are delivered to receivers that do not need it, and some local filtering mechanism is needed. In Figure 3 we can see how the model changes for applications that use end-to-end filtering. In the extreme case, one communication group is used for the entire session. Mapping and routing drop out as data is transmitted to all receivers in the session.

Figure 4 illustrates the mappings between the application and the network when an addressable routing architecture is offered.

[3]Note we have skipped over transport-level services, which, in a traditional, layered architecture, could easily also make use of this mapping; alternatively, transport-level functions are integrated into the application level with the ALF approach.

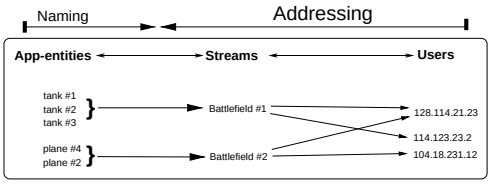


Fig. 4. Application to network mapping when over an addressable routing architecture.

In contrast to the previous two cases, the routers aid the application in the mapping data flows to addressable subsets of hosts.

Traditional IP multicast is based on a one-to-one mapping of data flows to multicast groups, establishing an implicit filtering by routers. Hence, for hosts not to receive data they do not need, a separate multicast group must be established for each set of application entities that are of interest to any set of hosts. A more scalable approach consists of mapping sets of data flows to a communication group connecting a common set of hosts, and allowing routers to filter dynamically those streams that are of interest to different subsets of hosts. In terms of naming, what Figure 4 suggests is a generalization to ALF, in which applications and the network interact to attain interest-based routing of information.

IV. Modeling Group Communication

In this section, we review the problem of choosing between filtering and addressing. We introduce a simulation model of the problem on which we elaborate some results. The model is deliberately simple and has been chosen to highlight specific behaviors. As using either filtering, addressing, or both will route data correctly to receivers, our criteria for evaluating the best choice is based on how each mechanism affects application performance and network resources. A discussion of implications of our results to the complicated multicast architecture is discussed at length in Section VI.

A. Modeling Application Types

We modeled two general application types: those with short-lived *transaction* data flows and those with long-lived, or *long duration* data flows. For example, a salient characteristic of DIS applications is the short duration of some data flows, e.g., fast planes or missiles, or the time one stays in a room. An example long-duration application is USENET newsreaders and news distribution. We can classify USENET as long duration because the data flows, which can be either articles or newsgroups, are relatively long. In other words, a summary description of the article's content, perhaps the subject and the author's name, is much shorter than the article's full text.

Certainly, most applications do not fit either of these two types exactly, and are likely to have aspects of both; but, it is easier to study a range of applications by characterizing the endpoints. We are not attempting to map out a complete model of all applications by developing a complex model, but rather to illuminate some important issues using a simpler model.

B. Modeling Network Support

Our performance simulation, described in detail shortly, modeled three network support scenarios highlighted in Section II:

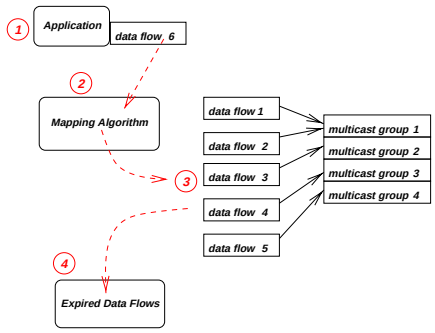


Fig. 5. The simulation model.

- One multicast group for all flows, corresponding to the filtering approach.
- One multicast group for each data flow, corresponding to the addressing approach.
- A hybrid-scenario, where there are less multicast groups than data flows, corresponding to the combined use of filtering and addressing. This approach makes sense because of the scarcity of class D addresses and the overhead of setting up many multicast groups.

The goal of the simulation was to quantify the overhead of each scenario by counting:

- The amount of *useful data* delivered to interested receivers, i.e., interesting data.
- The amount of *superfluous data* delivered to uninterested receivers, i.e., uninteresting data.
- The amount of *control traffic* generated by the scenario, caused by subscription and unsubscription to multicast groups.

These values are dependent on whether transaction or long-duration data flows are present.

C. Assumptions

Figure 5 depicts the simulation methodology. The simulation considered a single application that generated requests for data flows (Step 1). Each request had as parameters H, a randomly chosen subset of hosts, between two and seven out of 30 from a uniform distribution, and D, the duration of the data flow in simulation time units. The uniform distribution represents an application where content is uniformly interesting to a low number of hosts in the session, never favoring one particular set of hosts. The results will vary for different interest distributions, but the insights are gained from this experiment are generally applicable to any situation where there exists data that is not interesting to all receivers. For long-duration data flows, D is chosen from a geometric distribution with a given mean d, and minimum value of 1. A geometric distribution is not meant to be representative of all applications; the results of the simulation hold true because of the disparity of receiver interest, not because of the distribution of the inter-arrival of data flows. For transaction data flows, $D = 1$ for all flows. The time I, at which the next data flow arrives, is chosen at this time from a geometric distribution with a given mean i. When $I = 0$, an additional data flow arrives in the same time unit. When new requests arrive before existing requests expire, or when multiple requests arrive together, then *concurrent requests* existed in the simulation.

We defined a transaction application as one with short-lived data flows where d was small, and a long duration application as one with long-lived data flows where d was large. In order to keep the amount of data roughly the same between the two, long-duration applications had a larger value for i, and transaction applications had smaller value for i. Accordingly, in the simulations, the values of d and i determined whether the simulation was modeling a transaction application or a long-duration application.

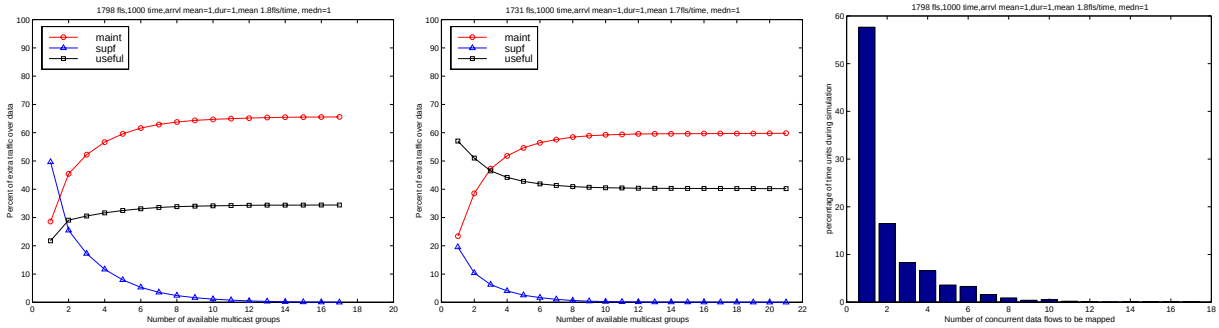
Each request generated is considered by a mapping algorithm (Step 2) that then associates the data flow with a multicast group (Step 3). The maximum number of concurrent multicast groups available during each simulation run is set to n. When $n = 1$, the simulation modeled filtering; when $n = \infty$, the simulation modeled addressing; and when n was less then the maximum number of concurrent requests generated throughout the simulation, a hybrid scheme was modeled. Once a data flow is mapped to an available multicast group, one unit of useful data is counted per time unit for each host interested in the data flow.

If no multicast group address is free at the start of a new flow, then more than one data flow is mapped to a single, existing multicast group. When this occurs, it is likely that the multiple data flows do not have completely matching sets of interested hosts; therefore, superfluous traffic, 1 unit per time unit, is counted at the uninterested hosts receiving data from another data flow mapped to the same multicast group. After a time, data flows expire (Step 4) and are removed from the mapping table.

Among mappings searched, the mapping algorithm picks the one mapping of data flows to the communication groups resulting in the least amount of superfluous traffic. Data flow expirations also caused re-mapping so that the superfluous traffic is always kept as low as possible.

When a host is required to join the multicast group to which a data flow is mapped, a unit of "control traffic" is counted, unless the host is already subscribed. One unit of control traffic is also counted when a data flow is re-mapped to a different multicast group (that the host does not already subscribe to), and one unit is counted for unsubscribing from a group. Counting control traffic in this fashion provided a crude approximation of what signaling would actually occur between routers, as if the multicast routing tree had a star topology. Note the costs of a unit of control traffic and data traffic are equal when determining the total traffic. However, if two mappings resulted in the same amount of superfluous traffic, then the mapping with the least amount of resultant control traffic was deemed a better choice.

We do not discuss the design of a specific mapping algorithm, or the communication between hosts required to perform the mapping. The simulation assumes a no-cost mapping and communication algorithms to measure the maximum performance possible under each scenario. At first, we performed the simulations using a mapping algorithm that compared all possible mappings and always found the mapping with the least amount of superfluous traffic. Unfortunately, the number of possible mappings is combinatorially explosive because it is a function of the number of data flows and the number of available multicast groups. Simulation times became intractable. Therefore, the results in this paper are based on an approximate mapping algorithm. For d data flows and c communication groups, the binary algorithm found the best mapping of the d data flows when



(a) $2 \leq H \leq 8$ hosts of 30 (b) $H \geq 20$ hosts of 30 (c) Distribution of concurrent data flows

Fig. 6. Transaction data flows.

$c = 2$. For $c > 2$, the group with the larger amount of superfluous traffic was spilt into two groups again. Next, the communication group with the largest amount of the superfluous traffic among the three was spilt, and so on. Simple test cases were used to show that our binary mapping algorithm performed relatively similar to combinatorially explosive algorithm, but within reasonable execution times. The tests we ran also showed that the mapping algorithm we used always performed better than random mapping. As we discuss subsequently, our results suggest the hybrid approach and the design of this kind of difficult mapping algorithm [30] should be avoided by instead designing a supportive multicast architecture.

V. Performance Results

We ran two major sets of simulations: one set for transaction applications, and one set for long-duration applications. Within each set, the same parameters were used, and thus, the same sequence of data flows were generated. To consider the three network support scenarios for each set, the number of available multicast groups was increased by one for each run, starting from one. Each iteration was run for 1000 simulation time units, at which point the results converged.

Figure 6a shows the percentage of each type of traffic for transaction data flow applications when the simulation is run over a varying number of available multicast groups. The x-axis represents n, the number of available multicast groups. Each of the three types of traffic is responsible for a certain percentage of the total traffic during the simulation, and this result is plotted along the y-axis. *Superfluous data* is marked with "△" symbols, *useful data* with "▫" symbols, and *control traffic* with "○" symbols. For example, when two multicast groups are available, 25% of the traffic was superfluous data sent to hosts, 29% of the traffic was useful data sent to hosts, and 45% of the traffic was due to hosts subscribing or unsubscribing from the multicast groups.

A. Transaction applications

During the simulation of transaction applications, each data flow that was generated was assigned between two and eight hosts out of 30 that found the data flow interesting. Figure 6c shows the distribution of concurrent data flows that existed during each time unit as a percentage of the total duration of the simulation.

As can be expected, Figure 6a shows that addressing is more efficient than filtering for transaction applications. The model suggests that these applications will suffer in performance due to the excessive control traffic resulting from setting up groups so that data is scoped correctly. The superfluous traffic is not nearly as significant as control traffic, but a reduction in superfluous traffic is seen with an increased number of multicast groups. Accordingly, for such applications, the control overhead must somehow be minimized even when multicast groups are not considered a scarce resource.

This analysis has two caveats. First, the cost of a unit of control traffic and a unit of data is considered the same. If units of data traffic where considered to be worth some multiple of control traffic, then the the percentage of control traffic would be lower in the graph. However for transaction data flows, an equal value is reasonable. Second, the apparent asymptotic nature of the plots can be explained by the distribution of the number of concurrent data flows to be mapped by the service. In the simulations, higher numbers of concurrent data flows are less likely, and adding additional multicast groups is beneficial a decreasing percentage of the time.

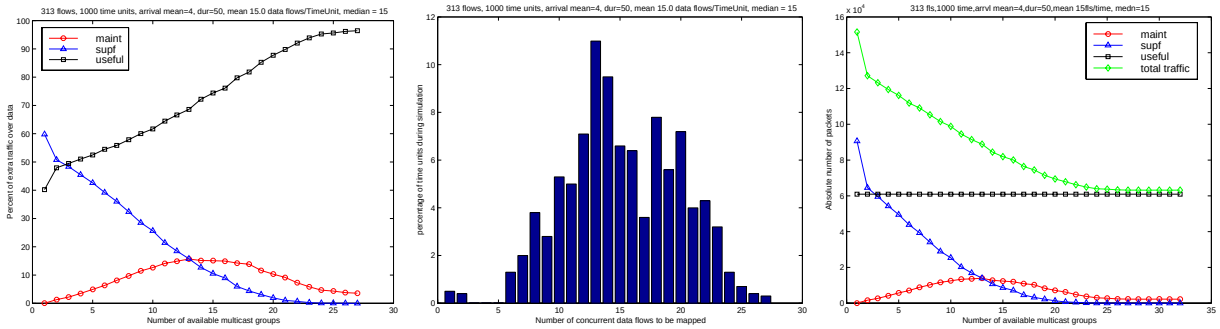
Figure 6b illustrates the importance of the interest-level of the data for the receiver set. For this alternate set of simulation runs, all generated data flows, H always contained between 20–30 hosts out of 30. When more receivers are interested in the transmitted data, then fewer receivers have to process superfluous data.

B. Long-lived data flow applications

Figure 7a shows the performance results for long-duration data flow applications. In this set of simulation runs, $i = 4.0$ time units and $d = 50.0$ time units. Figure 7b shows the associated distribution of concurrent data flows per time unit.

Figure 7a illustrates that long-lived data flow applications benefit greatly from each separate multicast group that is available: filtering does not provide the efficiency of addressing. Figure 7c illustrates the amount of traffic during the simulation in absolute values. It is clear that the total amount of traffic in the network reduces greatly as the number of multicast groups increases.

Figure 7a indicates that applications spend the least amount



(a) $2 \leq H \leq 8$ of 30 (b) Distribution of concurrent flows (c) Absolute values of traffic

Fig. 7. Long-duration data flows.

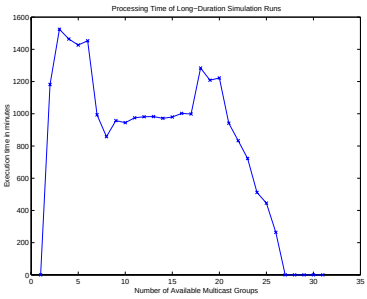


Fig. 8. Execution times.

of time remapping data flows into groups for pure filtering and pure addressing. The curve of control traffic is explained by the distribution of data flows. As the number of available multicast groups increases from a small number, the hosts spend more time using the groups to lower superfluous traffic. When the number of multicast groups is larger than the median number of concurrent data flows, most of the time each data flow has its own group. A reduction in control traffic occurs because often only a single join and single quit are needed, rather than a remapping because data flows expire or are created.

Figure 7 shows the execution time of the simulation for each iteration of available multicast groups, which demonstrates the difficulty of mapping flows when a hybrid of filtering and addressing is used. The amount of processing to determine the mapping severely drops off as the number of available groups approaches the maximum number of concurrent groups, or when simple filtering is used.

VI. Support From the Internet Architecture

In Section II we have shown there are a number of characteristics that differentiate large-scale applications. In the previous section, we have shown by simulation the following characteristics for generic application types:

- Even a limited hybrid scheme has better performance than filtering.
- Hybrid schemes are only an advantage over addressing when addresses are a scarce resource and mapping mechanisms are efficient.
- Addressing makes the most efficient use of network resources.
- Addressing is only feasible when addresses are not a scarce resource.
- Addressing must be lightweight, requiring low-overhead methods of setting up and tearing down groups and routes.

We can also state the following for the applications and protocols studied in Section II:

- Such applications require low latency address allocation.
- Addressing mechanisms must have low latency route setup and tear-down.
- New multicast groups and routes must be easily announced for receiver-driven joins, or easily configurable for sender-initiated schemes.
- Receiver set heterogeneity and partitioning information must be available to higher-level protocols when such requirements are performance based.

For those reasons, we believe the most promising long-term solution to this problem is to reconsider the multicast architecture to better support network layer addressing. Internet-wide, large-scale applications are still a research topic, allowing the opportunity to consider longer-term solutions. In this section, we consider the properties of IP multicast and show where changes can be made to support large-scale applications.

The above characteristics are in conflict with the current IP service model and architecture, which offers addressing only in the form of multiple multicast groups. IP multicast is a transport-level addressing service and with poor group management. There are four major problems with support in IP multicast: group set up latency, overhead, scarcity of addresses, and network-layer cooperation.

A. Setup Overhead

Based on current IETF standards, a common IP multicast architecture includes IGMP, PIM-SM, and MSDP; in the future a combination of MAAA, IGMP, PIM-SM, and BGMP will be more common. Alternatives to PIM include MOSPF and DVMRP. Setting up new multicast groups in IP requires a number of steps. First, a class-D address is chosen randomly by the source. Next, hosts join the group by using IGMP to signal their local router. The routers then send join messages hop-by-hop to the domain's PIM-SM rendezvous-point. Hosts in the same domain as the source learn of the new source from packets sent over the RP's shared tree they have already joined. They join directly on the shortest path to the router of the new source. RPs use MSDP to announce the group to all other RPs in remote do-

mains. Remote RPs in domains with hosts joined to the group then form a tunnel through the Internet to the source's router.

In the planned architecture, source's request a group address with the assistance of the Multicast Address Allocation Architecture [31], which includes the MADCAP, AAP, and MASC protocols. Also, MSDP will be replaced by BGMP, which offers shared bidirectional shared trees rather than dynamic tunnels between domains.

The signaling in these protocols does not allow for blocks of addresses to be set up at one time, except in the case of MASC where blocks of addresses can be allocated to domains. In contrast, network-level approaches make use of an existing multicast group to serve a subset of receivers, which requires less overhead and protocol steps to configure the new subgroup. Often groups can be torn down as data is sent [19].

B. Group Setup Latency

The protocols that are required to setup a multicast that we have listed above cost time as well as signaling. With the current architecture, group setup latency is dependent on the speed of the underlying multicast routing tree protocol. When an explicit join protocol like CBT [32] or BGMP [33] is used to construct the multicast spanning tree, sources must wait to send data until all receivers join the tree. When broadcast & prune multicast routing protocols like DVMRP [34] or PIM-DM [35] are used, the spanning tree is constructed as the data is sent. But, broadcast & prune routing protocols effectively turn transport-level addressing schemes into filtering schemes for brief periods as all routers in the domain are required to prune back unwanted groups. Thus, there is a tradeoff between the two multicast routing tree approaches: explicit-join protocols construct trees with less traffic, but have a longer latency. Whatever the needs of users' applications, the tradeoff is left to the control of system administrators as only one approach is used per domain.

Delays may also be caused by the operation of MAAA protocols, or the peer-by-peer signalling of MSDP.

In network-level addressing schemes, there is an opportunity to choose the method of construction of addressable multicast subgroups independent of the protocol used to construct the main multicast routing tree. For example, PGM and LWM create addressable subgroups without broadcasting, even if the underlying multicast routing protocol uses broadcasting. In AIM, subset multicast groups can be constructed by broadcasting data within the original multicast tree for low latency or by explicit join mechanisms within the original spanning tree for efficient construction traffic. PGM groups can be torn down as data traverses routers. Only the construction of the main spanning tree in AIM, PGM, and LWM is dependent on the underlying multicast routing protocol used. Lightweight subgroups are constructed without the use of MAAA, MSDP, BGMP, or other interdomain routing requirements.

C. Network information

Network information is generally not passed upward in IP. Although several of the applications protocols we have examined require partitioning of the receiver set, and those partitions are based on performance data, acquiring or collecting such information in IP is difficult. For example, Tracer makes use of MTRACE functionality, even though it was intended for debugging purposes.

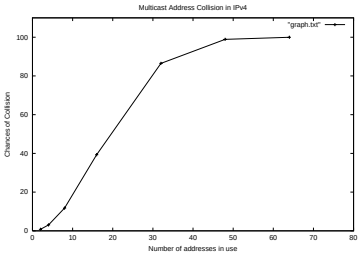


Fig. 9. The chances of collision in the IPv4 address space.

The underlying architecture cannot be modified every time a new partitioning scheme is desired. However, large-scale multicast applications would be better served by passing more information up to higher levels, such as topology or latency information. Passing information to higher levels does not violate separation of function or the end-to-end argument. For example, AIM passes up topology information to hosts via positional labels, which are labels that manifest the location of hosts relative to the source or core of the multicast tree.

D. Address Space

Proposals exist for mapping data flows to a smaller set of multicast addresses, but currently multicast addresses are not a scare resource. This is mainly due to a lack of commercial deployment of multicast [36].

Large-scale applications may be supported by numerous multicast groups for each flow of content, possibly as short-lived flows numbering in the hundreds or thousands. Increasing the use of multicast addresses drastically increases the probability of address *collision* across domains when the choices are random. Figure 9 shows a the chances of a collision given a number of randomly chosen addresses [36].

The set of MAAA protocols are still under study and are not a deployed standard. Alternative solutions to address space problems include address space available with IPv6, the (host,group) source-rooted paradigm of Express [37], or (core,group) bidirectional shared trees of Simple Multicast [38]. PGM and AIM assumes that a multicast group is an interdomain *context* that many hosts share. Addressable subgroups in PGM and AIM use a (group,stream) paradigm so that subgroup addresses are plentiful within the multicast group context and also offer hints for aggregation.

D.1 Multicast Routing Trees

Some multicast routing protocols offer *per-source* routing trees and others *shared* routing trees. With per-source trees, each packet is delivered on the shortest path from each source to the receiver set. Shared routing trees use a common routing tree for all sources: either a bidirectional tree, or a unidirectional tree is used. In the former case, packets are disseminated from any point on the tree; in the later case, packets are first unicast to a rendezvous point, which then multicasts the packet to the receiver set. As such, per-source tree incur source·group state at routers, and shared trees incur group state only.

Often it is assumed that shared trees are better for large-scale applications, such as DIS, because each participant is usually a receiver and a source of data. However, if the content is not of general interest, then shared trees result in an inefficient approach. It is likely that multiple shared trees will have to be setup, causing per-data flow state in the network for multiple sources. What is needed instead are shared trees that allow efficient subgroup creation to allow addressing.

VII. Conclusions

We have examined the requirements and performance of large-scale applications and related protocols and shown that they are incongruent with IP multicast. When receiver interest in content is not considered for applications, then support using broadcast and filter methodsm, e.g., in USENET or DIS, is grossly inefficient. Hybrid approaches are only an advantage when multicast addresses are a scarce resource and an efficient mapping algorithm is designed. The difficulty of discovering performance characteristics and small address space of the routing architectures of IP multicast do not match well with large-scale applications. Because IP multicast groups are subject to large setup overhead and long setup latency, including a long address allocation procedure, we propose reconsidering the architecture to support a lightweight network-level addressing approach.

There are research initiatives that better position IP multicast to support large-scale applications, but that require the IP multicast model to be modified. PGM allows data filtering and subgrouping in routers. AIM offers per-packet multicast groups for small sets of receivers, extensible anycasting, and flexible subgroup construction. Express and SM offer an addressing scheme that would solve problems in group management and the scarcity of multicast addresses.

Our future work is to take these characteristics as requirements of designs of future addressing architectures that must support these applications and related protocols. Additionally, an application-level mechanism to announce naming structures and content and for receivers to express interest is required by this work.

References

[1] C. Diot and L. Gautier, "A distributed architecture for multiplayer interactive applications on the internet," *IEEE Networks magazine*, vol. 13, no. 4, July/August 1999.

[2] A. Pang and C. Wittenbrink, "Collaborative 3D visualization with CSpray," *IEEE Comp. Graphics and Aplications*, vol. 17, no. 2, pp. 32–41, March 1997.

[3] M. Pullen, M. Myjak, and C. Bouwens, "Limitations of internet protocol suite for distributed simulation in the large multicast environment," Tech. Rep., Request for Comments 2502, 1999.

[4] S. Bhattacharyya, J. Kurose, D. Towsley, and R. Nagarajan, "Efficient rate-controlled bulk data transfer using multiple multicast groups," in *Proc. IEEE Infocom98*, April 1998, pp. 1172–9.

[5] D. De Lucia and K. Obraczka, "Multicast feedback suppresion," in *Proc. IEEE INFOCOM'97*. IEEE, 1997, pp. 463–70.

[6] S. Kasera, J. Kurose, and D. Towsley, "Scalable reliable multicast using multiple multicast groups," in *ACM Sigmetrics*, June 1997.

[7] S. Paul, K. K. Sabnani, J. C. Lin, and S. Bhattacharyya, "Reliable multicast transport protocol (RMTP)," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 407–421, April 1997.

[8] J. Pullen and V.P. Laviano, "A selectively reliable transport protocol for distributed interactive simulation," in *Proc. 13th DIS Workshop on Standards for the Interoperability of Distributed Simulations*, September 95.

[9] C.K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," in *Proc. ACM SIGCOMM*, September 1998, pp. 68–79.

[10] B. Kantor and P. Lapsley, "Network news transfer protocol," Request for Comments 977, Feb 1986.

[11] Logs available from http://www.cs.umass.edu/ brian/usenet.

[12] K. Morse, "Interest management in large scale distributed simulations," Tech. Rep., UCI Technical Report, 1996, 96-27.

[13] M. Macedonia, M. Zyda, D. Pratt, D. Brutzman, and P. Barham, "Exploiting reality with multicast groups," *IEEE Computer Graphics and Applications*, vol. 15, no. 5, September 1995.

[14] J. Calvin, C. Chiang, and D. VanHook, "Data subscription," in *Proc. of 12th DIS workshop*, March 1995.

[15] S. Rak and D. Van Hook, "Evaluation of grid-based relevance filtering for multicast group assignment," in *Proc. of 14th DIS workshop*, March 1996.

[16] E. Léty and T. Turletti, "Issues in designing a communication architecture for large-scale virtual environments," in *Proceedings of the 1st International Workshop on Networked Group Communication*, November 1999.

[17] B. N. Levine, S. Paul, and J.J. Garcia-Luna-Aceves, "Organizing multicast receivers deterministically according to packet-loss correlation," in *Proc. Sixth ACM International Multimedia Conference (ACM Multimedia 98)*, September 1998.

[18] B. N. Levine and J.J. Garcia-Luna-Aceves, "Improving internet multicast with routing labels," in *Proc. IEEE International Conference on Network Protocols*, October 1997, pp. 241–50.

[19] T. Speakman, D. Farinacci, S. Lin, and A. Tweedly, "Pragmatic general multicast," Tech. Rep., internet draft, January 1998.

[20] D. Li and D. R. Cheriton, "OTERS (On-Tree Efficient Recovery using Subcasting): A reliable multicast protocol," in *Proc. IEEE International Conference on Network Protocols (ICNP'98)*, October 1998, pp. 237–245.

[21] S. Ratnasamy and S. McCanne, "Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements," in *Proceedings of IEEE INFOCOM'99*, March 1999.

[22] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. ACM SIGCOMM*, August 1996, pp. 117–130.

[23] D.D. Clark and D.L. Tennenhouse, "Architectural considerations for a new generation of protocols," in *SIGCOMM'90*. ACM, September 1990, pp. 200–208.

[24] S. Raman and S. R. McCanne, "Generalized data naming and scalable state announcements for reliable multicast," Tech. Rep. UCB/CSD-97-951, UC Berekely, June 1997.

[25] M. Fuchs, C. Diot, T. Turletti, and M. Hoffman, "A naming approach for ALF design," in *Proc. HIPPARCH'98 workshop*, UCL London, UK, June 1998.

[26] J.F. Shoch, "Inter-networking naming, addressing & routing," in *Proceedings of IEEE COMPCON*, fall 1978, pp. 72–79.

[27] C. Papadopoulos, G. Parulkar, and G. Varghese, "An error control scheme for large-scale multicast applications," in *Proc. IEEE INFOCOM'98*, 1998, pp. 1118–1196.

[28] Tibco Inc., "Tib/rendezvous," white paper, Tibco, 1998, http://www.rv.tibco.com/rvwhitepaper.html.

[29] J. Randell and M. Wray, "The aurora event distribution system," Tech. Rep., Hewlett-Packard Laboratories, Bristol, UK, January 1997.

[30] T. Wong, R. Katz, and S. McCanne, "A preference clustering protocol for large-scale multicast applications," in *Proceedings of the First International Workshop on Networked Group Communication*, November 1999.

[31] M. Handley, D. Thaler, and D. Estrin, "The internet multicast address allocation architecture," Tech. Rep. draft-handley-malloc-arch-∗.ps, IETF Internet draft, Dec 1997.

[32] A. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (CBT): An architecture for scalable inter-domain multicast routing," in *Proc. ACM SIGCOMM'93*, October 1993, pp. 85–95.

[33] K. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, and M. Handley, "The MASC/BGMP architecture for inter-domain multicast routing," in *Proc. ACM SIGCOMM 98*, September 1998.

[34] D. Waitzman, C. Partridge, and S. Deering, "Distance vector multicast routing protocol," Request for Comments 1075, November 1988.

[35] S. Deering et al., "An architecture for wide-area multicast routing," in *Proc. ACM SIGCOMM'94*, 1994, pp. 126–135.

[36] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the ip multicast service and architecture," *IEEE Network magazine special issue on Multicasting*, January/February 2000.

[37] Hugh W. Holbrook and David R. Cheriton, "Ip multicast channels: Express support for large-scale single-source applications," in *Proc. ACM SIGCOMM'99*, September 1999.

[38] T. Ballardie, J. Crowcroft, C. Diot, C.-Y. Lee, R. Perlman, and Z. Wang, "On extending the standard ip multicast architecture," Tech. Rep. RN/99/21, UCL, October 1999.